Bayesian Nonparametrics:

DPMM

+  

Indian Buffet Process
Reminders

• Cloud Credits (AWS or GCP)
  – first request deadline: Thu at 11:59pm

• Quiz 3
  – Mon, May 3 during lecture slot
  – Topics: Lectures 16 - 23
Exchangability

Question:
Select All: Which of the following properties of an infinite sequence of random variables \(X_1, X_2, X_3, \ldots\) ensure that they are infinitely exchangeable?

- For any pair of orderings \((i_1, i_2, \ldots, i_n)\) and \((j_1, j_2, \ldots, j_n)\) of the indices \((1, \ldots, n)\) the joint probability of the two orderings is the same
- The joint distribution is invariant to permutation
- The joint distribution of the first \(n\) random variables can be represented as a mixture
- The random variables are independent and identically distributed

Answer:
Chinese Restaurant Process & Stick-breaking Constructions

DIRICHLET PROCESS MIXTURE MODEL
CRP Mixture Model

• Draw $n$ cluster indices from a CRP:
  
  $z_1, z_2, \ldots, z_n \sim CRP(\alpha)$

• For each of the resulting $K$ clusters:
  
  $\theta_{k^*} \sim H$

  where $H$ is a base distribution

• Draw $n$ observations:
  
  $x_i \sim p(x_i \mid \theta_{z_i}^*)$

Customer $i$ orders a dish $x_i$ (observation) from a table-specific distribution over dishes $\theta_{k^*}$ (cluster parameters)

(color denotes different values of $x_i$)
CRP Mixture Model

- Draw n cluster indices from a CRP:
  \[ z_1, z_2, \ldots, z_n \sim CRP(\alpha) \]
- For each of the resulting K clusters:
  \[ \theta_k^* \sim H \]
  where \( H \) is a base distribution
- Draw n observations:
  \[ x_i \sim p(x_i | \theta_{z_i}^*) \]

- The Gibbs sampler is easy thanks to exchangeability
- For each observation, we remove the customer / dish from the restaurant and resample as if they were the last to enter
- If we collapse out the parameters, the Gibbs sampler draws from the conditionals:
  \[ z_i \sim p(z_i | z_{-i}, x) \]

(color denotes different values of \( x_i \))
CRP Mixture Model

Overview of 3 Gibbs Samplers for Conjugate Priors

• Alg. 1: (uncollapsed)
  – Markov chain state: per-customer parameters $\theta_1, \ldots, \theta_n$
  – For $i = 1, \ldots, n$: Draw $\theta_i \sim p(\theta_i \mid \theta_{-i}, x)$

• Alg. 2: (uncollapsed)
  – Markov chain state: per-customer cluster indices $z_1, \ldots, z_n$ and per-cluster parameters $\theta_1^*, \ldots, \theta_k^*$
  – For $i = 1, \ldots, n$: Draw $z_i \sim p(z_i \mid z_{-i}, x, \theta^*)$
  – Set $K = \text{number of clusters in } z$
  – For $k = 1, \ldots, K$: Draw $\theta_k^* \sim p(\theta_k^* \mid \{x_i : z_i = k\})$

• Alg. 3: (collapsed)
  – Markov chain state: per-customer cluster indices $z_1, \ldots, z_n$
  – For $i = 1, \ldots, n$: Draw $z_i \sim p(z_i \mid z_{-i}, x)$
CRP Mixture Model

• Q: How can the Alg. 2 Gibbs samplers permit an infinite set of clusters in finite space?
• A: Easy!
  – We are only representing a finite number of clusters at a time – those to which the data have been assigned
  – We can always bring back the parameters for the “next unoccupied table” if we need them
Dirichlet Process Mixture Model

*Whiteboard*

– Dirichlet Process Mixture Model
  (stick-breaking version)
CRP-MM vs. DP-MM

Dirichlet Process: For both the CRP and stick-breaking constructions, if we marginalize out $G$, we have the following predictive distribution:

$$
\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{1}{\alpha + n} \left( \alpha H + \sum_{i=1}^{n} \delta_{\theta_i} \right)
$$

(Blackwell-MacQueen Urn Scheme)

The Chinese Restaurant Process Mixture Model is just a different construction of the Dirichlet Process Mixture Model where we have marginalized out $G$. 

$$
\begin{align*}
\theta_{n+1}|\theta_1, \ldots, \theta_n & \sim \frac{1}{\alpha + n} \left( \alpha H + \sum_{i=1}^{n} \delta_{\theta_i} \right) \\
& = \frac{1}{\alpha + n} \left( \alpha \sum_{i=1}^{n} \delta_{\theta_i} + \sum_{i=1}^{n} \delta_{\theta_i} \right) \\
& = \frac{1}{\alpha + n} \left( \alpha H + \sum_{i=1}^{n} \delta_{\theta_i} \right)
\end{align*}
$$
Graphical Models for DPMMs

The Pólya urn construction

The Stick-breaking construction
Example: DP Gaussian Mixture Model

Figure from Blei & Jordan (2006)

Figure 2: The approximate predictive distribution given by variational inference at different stages of the algorithm. The data are 100 points generated by a Gaussian DP mixture model with fixed diagonal covariance.
Example: DP Gaussian Mixture Model

Figure 3: Mean convergence time and standard error across ten data sets per dimension for variational inference, TDP Gibbs sampling, and the collapsed Gibbs sampler.
Summary of DP and DP-MM

• **DP** has many **different representations:**
  – Chinese Restaurant Process
  – Stick-breaking construction
  – Blackwell-MacQueen Urn Scheme
  – Limit of finite mixtures
  – etc.

• These representations give rise to a variety of **inference techniques** for the **DP-MM** and related models
  – Gibbs sampler (CRP)
  – Gibbs sampler (stick-breaking)
  – Variational inference (stick-breaking)
  – etc.
GMM VS. DPMM EXAMPLE
Example: Dataset
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=0)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=5)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=10)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=15)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=20)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=25)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=30)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=35)
Example: GMM

Clustering with GMM (k=6, init=random, cov=full, iter=39)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=0)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=1)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=2)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=3)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=4)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=5)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=6)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=7)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=8)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=9)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=10)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=11)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=12)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=13)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=14)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=15)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=16)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=17)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=18)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=19)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=20)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=21)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=22)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=23)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=24)
Example: DPMM

Clustering with DPMM (k=6, init=random, cov=full, iter=25)
HIERARCHICAL DIRICHLET PROCESS (HDP)
Related Models

- Hierarchical Dirichlet Process Mixture Model (HDP-MM)
- Infinite HMM
- Infinite PCFG
HDP-MM

- In LDA, we have \( M \) independent samples from a Dirichlet distribution.
- The weights are different, but the topics are fixed to be the same.
- If we replace the Dirichlet distributions with Dirichlet processes, each atom of each Dirichlet process will pick a topic \textit{independently} of the other topics.
- Because the base measure is \textit{continuous}, we have zero probability of picking the same topic twice.
- If we want to pick the same topic twice, we need to use a \textit{discrete} base measure.
- For example, if we chose the base measure to be
  \[
  H = \sum_{k=1}^{K} \alpha_k \delta_{\beta_k}
  \]
  then we would have LDA again.
- We want there to be an infinite number of topics, so we want an \textit{infinite, discrete} base measure.
- We want the location of the topics to be random, so we want an \textit{infinite, discrete, random} base measure.
Hierarchical Dirichlet process:

\[ G_0 \mid \gamma, H \sim \text{DP}(\gamma, H) \]
\[ G_j \mid \alpha, G_0 \sim \text{DP}(\alpha, G_0) \]
\[ \theta_{ji} \mid G_j \sim G_j \]

Figure from Teh MLSS 2007
Figure 6: (Left) Comparison of latent Dirichlet allocation and the hierarchical Dirichlet process mixture. Results are averaged over 10 runs; the error bars are one standard error. (Right) Histogram of the number of topics for the hierarchical Dirichlet process mixture over 100 posterior samples.
HDP-HMM (Infinite HMM)

Number of hidden states in Infinite HMM is countably infinite

Figure 9: A hierarchical Bayesian model for the infinite hidden Markov model.

Figure 10: Comparing the infinite hidden Markov model (solid horizontal line) with ML, MAP and VB trained hidden Markov models. The error bars represent one standard error (those for the HDP-HMM are too small to see).
HDP-PCFG (Infinite PCFG)

HDP-PCFG

\[ \begin{align*}
\beta & \sim \text{GEM}(\alpha) \quad \text{[draw top-level symbol weights]} \\
\text{For each grammar symbol } & z \in \{1, 2, \ldots \} : \\
\phi_z^T & \sim \text{Dirichlet}(\alpha^T) \quad \text{[draw rule type parameters]} \\
\phi_z^E & \sim \text{Dirichlet}(\alpha^E) \quad \text{[draw emission parameters]} \\
\phi_z^B & \sim \text{DP}(\alpha^B, \beta \beta^T) \quad \text{[draw binary production parameters]} \\
\end{align*} \]

For each node \( i \) in the parse tree:

- \( t_i \sim \text{Multinomial}(\phi_z^T) \) \quad \text{[choose rule type]}
- If \( t_i = \text{EMISSION} \):
  - \( x_i \sim \text{Multinomial}(\phi_z^E) \) \quad \text{[emit terminal symbol]}
- If \( t_i = \text{BINARY-PRODUCTION} \):
  - \( (z_{L(i)}, z_{R(i)}) \sim \text{Multinomial}(\phi_z^B) \) \quad \text{[generate children symbols]}

\[ \begin{align*}
\beta & \sim \text{GEM}(\alpha) \\
\text{state} \\
\end{align*} \]

\[ \begin{align*}
\beta \beta^T & \quad \text{left child state} \\
\text{right child state} \\
\phi_z^B & \sim \text{DP}(\beta \beta^T) \\
\text{left child state} \\
\text{right child state} \\
\end{align*} \]
### Parametric vs. Nonparametric

<table>
<thead>
<tr>
<th>Type of Model</th>
<th>Parametric Example</th>
<th>Nonparametric Example</th>
<th>Construction #1</th>
<th>Construction #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>distribution over counts</td>
<td>Dirichlet-Multinomial Model</td>
<td>Dirichlet Process (DP)</td>
<td>Chinese Restaurant Process (CRP)</td>
<td>Stick-breaking construction</td>
</tr>
<tr>
<td>mixture</td>
<td>Gaussian Mixture Model (GMM)</td>
<td>Dirichlet Process Mixture Model (DPMM)</td>
<td>CRP Mixture Model</td>
<td>Stick-breaking construction</td>
</tr>
</tbody>
</table>
Summary of DP and DP-MM

• **DP** has many **different representations:**
  – Chinese Restaurant Process
  – Stick-breaking construction
  – Blackwell-MacQueen Urn Scheme
  – Limit of finite mixtures
  – etc.

• These representations give rise to a variety of **inference techniques** for the **DP-MM** and related models
  – Gibbs sampler (CRP)
  – Gibbs sampler (stick-breaking)
  – Variational inference (stick-breaking)
  – etc.
INDIAN BUFFET PROCESS (IBP)
Outline

• **Motivation:** *Infinite* Latent Feature Models
• **Finite Feature Model**
  – Beta-Bernoulli Model
  – Marginalized Beta-Bernoulli Model
  – Expected # of non-zeros
  – Taking the *Infinite* Limit
  – Left-ordered form (equivalence classes)
• **The Indian Buffet Process (IBP)**
  – Nonexchangeable IBP
  – Exchangeable IBP
  – Gibbs Sampling with Exchangeable IBP
• **IBP properties**
• **Applications**
• **Summary**
Motivation

❖ Latent Feature Models
  – Examples:
    • factor analysis
    • probabilistic PCA
    • cooperative vector quantization
    • sparse PCA

❖ Applications
  – object detection in images
  – choice behavior (i.e. option A over option B)
  – proteomics: modeling the functional interactions of proteins – which can belong to multiple complexes at the same time
  – collaborative filtering: modeling features of movie preferences (a la. Netflix challenge)
  – structure learning for graphical models (i.e. bipartite graphs)
Latent Feature Models

Let \( x_i \) be the \( i \)th data instance
\( f_i \) be its features

Define \( X = [x_1^T, x_2^T, \ldots, x_N^T] \)
\( F = [f_1^T, f_2^T, \ldots, f_N^T] \)

Model: \( p(X, F) = p(X|F)p(F) \)
Latent Feature Models

Decompose the feature matrix, $F$, into a sparse binary matrix, $Z$, and a value matrix, $V$.

\[ F = Z \otimes V \]

where $\otimes$ is the elementwise product

\[ z_{ij} \in \{0, 1\} \]

\[ v_{ij} \in \mathcal{R} \]
Latent Feature Models

Decompose the feature matrix, $F$, into a sparse binary matrix, $Z$, and a value matrix, $V$.

$$F = Z \otimes V$$

where $\otimes$ is the elementwise product

$$z_{ij} \in \{0, 1\}$$

$$v_{ij} \in \mathcal{R}$$

Model: $p(X, F) = p(X|F)p(F)$

$$= p(X|F)p(Z)p(V)$$

The IBP will provide $p(Z)$ for the case of infinite columns!
Finite Feature Model

**Beta-Bernoulli Model**

**Generative Story:**

- for each feature $k \in \{1, \ldots, K\}$:
  - $\pi_k \sim \text{Beta}(\frac{\alpha}{K}, 1)$ where $\alpha > 0$
  - for each object $i \in \{1, \ldots, N\}$:
    - $z_{ik} \sim \text{Bernoulli}(\pi_k)$
Finite Feature Model

Marginalized Beta-Bernoulli Model

Because of the conjugacy of the Beta and Bernoulli, we can analytically marginalize out the feature prevalence parameters, $\pi_k$.

$$P(Z) = \prod_{k=1}^{K} \int \left( \prod_{i=1}^{N} P(z_{ik} | \pi_k) \right) p(\pi_k) d\pi_k$$

$$= \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K}) \Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}.$$  

where $m_k = \sum_{i=1}^{N} z_{ik}$ is # features ON in column $k$,  

$\Gamma$ is the Gamma function.
Finite Feature Model

Expected # of non-zeroes

Generative Story:

- for each feature \( k \in \{1, \ldots, K\} \):
  - \( \pi_k \sim \text{Beta}(\frac{\alpha}{K}, 1) \) where \( \alpha > 0 \)
  - for each object \( i \in \{1, \ldots, N\} \):
    - \( z_{ik} \sim \text{Bernoulli}(\pi_k) \)

Recall: if \( X \sim \text{Beta}(r, s) \), then \( \mathbb{E}[X] = \frac{r}{r + s} \)

if \( Y \sim \text{Bernoulli}(p) \), then \( \mathbb{E}[Y] = p \)

\[
\mathbb{E}[z_{ik}] = \frac{\alpha}{K} \frac{1}{1 + \frac{\alpha}{K}} 
\]

\[
\Rightarrow \mathbb{E}[\mathbf{1}^T \mathbf{Z} \mathbf{1}] = \mathbb{E} \left[ \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \right] = \frac{N \alpha}{1 + \frac{\alpha}{K}} 
\]

What happens as \( K \rightarrow \infty \)?

So the expected number of non-zero entries in \( Z \) is \( \leq N\alpha \)
Finite Feature Model

Taking the Infinite Limit

\[
\lim_{K \to \infty} p(Z) = \lim_{K \to \infty} \prod_{k=1}^{K} \frac{\alpha k \Gamma(m_k + \frac{\alpha}{K}) \Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}.
\]

\[
= 0
\]

Problem: Every **matrix** has **zero** probability!
Finite Feature Model

Left-Ordered Form (lof)

Topic Modeling:

• Consider many samples of the $k^{th}$ topic from the Markov chain:

\[ \phi_k^{(1)}, \phi_k^{(2)}, \ldots, \phi_k^{(T)} \]

This topic will “drift” over time (e.g. from \{politics\} at time (t) to \{geology\} at time (t+m))

• Instead of averaging, it’s common to use a MAP estimate of the topics

• The order of the topics is **not important** to the model (i.e. the topics are not identifiable)
Finite Feature Model

*Left-Ordered Form (lof)*

Back to our model:

• Q: In a *latent* feature model, what’s the difference between feature $k=13$ and $k=27$?

• A: Nothing!

The use of left-ordered form *capitalizes* on the fact that *features are not identifiable* (i.e. order of features doesn’t matter to the model)
Finite Feature Model

Left-Ordered Form (lof)

Define the history of feature $k$ to be the magnitude of the binary value given by the column:

$$h_k = \sum_{i=1}^{N} 2^{(N-i)} z_{ik}$$

$K_h = \#$ of features with history $h$

$K_0 = \#$ of features with $m_k = 0$ (i.e. $h = 0$)

$$K_+ = \sum_{h=1}^{2^N-1} K_h, \# \text{ of features with non-zero history}$$

$\Rightarrow K = K_0 + K_+$

Define lof($Z$) to be sorted left-to-right by the history of each feature.
Finite Feature Model

Left-Ordered Form (lof)

Define $lof(Z)$ to be sorted left-to-right by the history of each feature.

Define equivalence class $[Z] = \{ Z' : lof(Z') = lof(Z) \}$

Cardinality of $[Z] = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!}$
Finite Feature Model

Taking the Infinite Limit

\[
\lim_{K \to \infty} p(\mathbf{Z}) = \lim_{K \to \infty} \prod_{k=1}^{K} \frac{\alpha \Gamma(m_k + \frac{\alpha}{K}) \Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}.
\]

= 0

Problem: Every matrix has zero probability!

\[
\lim_{K \to \infty} p([\mathbf{Z}]) = \lim_{K \to \infty} \frac{K!}{\prod_{h=0}^{2N-1} K_h!} p(\mathbf{Z})
\]

= \frac{\alpha^{K+}}{\prod_{h=1}^{2N-1} K_h!} \cdot \exp\{-\alpha H_N\} \cdot \prod_{k=1}^{K+} \frac{(N-m_k)! (m_k-1)!}{N!},

where \(H_N = \sum_{j=1}^{N} \frac{1}{j}\) is the \(N\)th harmonic number

Solution: Every equivalence class has non-zero probability!
The Indian Buffet Process

Imagine an Indian restaurant with a buffet containing an infinite # of dishes.

N customers make a plate by selecting dishes from the buffet:

- 1st customer:
  Starts at the left and selects a Poisson($\alpha$) number of dishes

- $i^{\text{th}}$ customer:
  1. Samples previously sampled dishes according to their popularity:
     (i.e. with prob. $m_k/i$, where $m_k$ is the # of previous customers who tried dish $k$)
  2. Then selects a Poisson($\alpha/i$) number of new dishes

Problem: the process is not exchangeable – dishes sampled as “new” depend on the customer order.
Imagine an Indian restaurant with a buffet containing an infinite # of dishes.

N customers make a plate by selecting dishes from the buffet:

- 1st customer:
  Starts at the left and selects a Poisson(α) number of dishes

- i\textsuperscript{th} customer:
  1. Makes a single decision for dishes with same history, h:
     (i.e. If there are \(K_h\) dishes w/history \(h\) sampled by \(m_h\) customers, then she samples a Binomial(\(m_h/l\), \(K_h\)) number starting at the left)
  2. Then selects a Poisson(\(\alpha/i\)) number of new dishes

This yields a lof matrix, \(Z\).

Does so with probability \(p([Z])!\)
The Indian Buffet Process

Example:

Figure 6: A binary matrix generated by the Indian buffet process with $\alpha = 10$. The order in which the customers make their choices. However, if we only pay attention to the lof-equivalence classes of the matrices generated by this process, we obtain the exchangeable distribution $P(\mathbf{Z})$ given by Equation 14:

$$
\prod_{N} \binom{K}{i} \frac{1}{i!} \prod_{N-1} \binom{K}{h} \frac{1}{h!}
$$

Matrices generated via this process map to the same left-ordered form, and $P(\mathbf{Z})$ is obtained by multiplying $P(\mathbf{Z})$ from Equation 15 by this quantity.

It is possible to define a similar sequential process that directly produces a distribution on lof-equivalence classes in which customers are exchangeable, but this requires more effort on the part of the customers. In the exchangeable Indian buffet process, the first customer samples a Poisson($\alpha$) number of dishes, moving from left to right. The $i$th customer moves along the buffet, and makes a single decision for each set of dishes with the same history. If there are $K_h$ dishes with history $h$, under which $m_h$ previous customers have sampled each of those dishes, then the customer samples a Binomial($m_h i, K_h$) number of those dishes, starting at the left. Having reached the end of all previous sampled dishes, the $i$th customer then tries a Poisson($\alpha_i$) number of new dishes. Attending to the history of the dishes and always sampling from the left guarantees that the resulting matrix is in left-ordered form, and it is easy to show that the matrices produced by this process have the same probability as the corresponding lof-equivalence classes under Equation 14.

4.5 A Distribution over Collections of Histories

In Section 4.2, we noted that lof-equivalence classes of binary matrices generated from assignment vectors correspond to partitions. Likewise, lof-equivalence classes of general binary matrices correspond to simple combinatorial structures: vectors of non-negative integers. Fixing some ordering of $N$ objects, a collection of feature histories on those objects can be represented by a frequency vector.
Gibbs Sampler for IBP

Consider a “prior only” sampler of $p(Z | \alpha)$

- For finite $K$:
  \[
P(z_{ik} = 1 | z_{-i,k}) = \int_0^1 P(z_{ik} | \pi_k) p(\pi_k | z_{-i,k}) d\pi_k
  \]
  \[
  = \frac{m_{-i,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}},
  \]
  where $z_{-i,k}$ is the $k$th column except row $i$,
  $m_{-i,k}$ is the # of rows w/feat. $k$ except $i$

- For infinite $K$:
  - The “Exchangeable IBP” is exchangeable!
  - Choose an order s.t. the $i$th customer was the last to enter (just like CRP sampler)
  - For any $k$ s.t. $m_{i,k} > 0$, resample:
    \[
P(z_{ik} = 1 | z_{-i,k}) = \frac{m_{-i,k}}{N},
    \]
    - Then draw a Poisson($\alpha/i$) # of new dishes.
Properties of the Indian buffet process

\[ P([Z] | \alpha) = \exp \left\{ -\alpha H_N \right\} \frac{\alpha^{K_+}}{\prod_{h>0} K_h!} \prod_{k \leq K_+} \frac{(N - m_k)!(m_k - 1)!}{N!} \]

Shown in (Griffiths and Ghahramani, 2005):

- It is infinitely exchangeable.
- The number of ones in each row is Poisson(\( \alpha \)).
- The expected total number of ones is \( \alpha N \).
- The number of nonzero columns grows as \( O(\alpha \log N) \).

Additional properties:

- Has a stick-breaking representation (Teh, Görür, Ghahramani, 2007)
- Can be interpreted using a Beta-Bernoulli process (Thibaux and Jordan, 2007)
Posterior Inference in IBPs

\[ P(Z, \alpha|X) \propto P(X|Z)P(Z|\alpha)P(\alpha) \]

Gibbs sampling: \[ P(\tilde{z}_{nk} = 1|Z_{-(nk)}, X, \alpha) \propto P(\tilde{z}_{nk} = 1|Z_{-(nk)}, \alpha)P(X|Z) \]

- If \( m_{-n,k} > 0 \), \[ P(\tilde{z}_{nk} = 1|z_{-n,k}) = \frac{m_{-n,k}}{N} \]
- For infinitely many \( k \) such that \( m_{-n,k} = 0 \): Metropolis steps with truncation* to sample from the number of new features for each object.
- If \( \alpha \) has a Gamma prior then the posterior is also Gamma \( \rightarrow \) Gibbs sample.

Conjugate sampler: assumes that \( P(X|Z) \) can be computed.

Non-conjugate sampler: \( P(X|Z) = \int P(X|Z, \theta)P(\theta)d\theta \) cannot be computed, requires sampling latent \( \theta \) as well (c.f. (Neal 2000) non-conjugate DPM samplers).

*Slice sampler: non-conjugate case, is not approximate, and has an adaptive truncation level using a stick-breaking construction of the IBP (Teh, et al, 2007).

Particle Filter: (Wood & Griffiths, 2007).

Accelerated Gibbs Sampling: maintaining a probability distribution over some of the variables (Doshi-Velez & Ghahramani, 2009).

Variational inference: (Doshi-Velez, Miller, van Gael, & Teh, 2009).
Latent variable model: let $X$ be the $N \times D$ matrix of observed data, and $Z$ be the $N \times K$ matrix of binary latent features

$$P(X, Z|\alpha) = P(X|Z)P(Z|\alpha)$$

By combining the IBP with different likelihood functions we can get different kinds of models:

- Models for graph structures (w/ Wood, Griffiths, 2006)
- Models for protein complexes (w/ Chu, Wild, 2006)
- Models for overlapping clusters (w/ Heller, 2007)
- Models for choice behaviour (Görür, Jäkel & Rasmussen, 2006)
- Models for users in collaborative filtering (w/ Meeds, Roweis, Neal, 2006)
- Sparse latent factor models (w/ Knowles, 2007)
Summary

• Beta-Bernoulli model is a simple finite feature model
• Can treat features as latent
• **Infinite limit** of Beta-Bernoulli yields the Indian Buffet Process (IBP)
• Many properties of the IBP are similar to the CRP